CHAPTER 1
INTRODUCTION

In the present day scenario, ever increasing security threats, diversifying compliance requirements, consolidation and cloud computing technology are just a few of the reasons to establish the fact that data security has become critical. Almost a decade after the first U.S. breach notification law, the need for strong preventive controls continues to increase as access to data expands. Stolen client devices, including tablets and smart phones have the potential to easily expose sensitive information as users move beyond the laptop. Outsourcing, off shoring, corporate mergers, and nearly continuous organizational change create additional risks by making it easier for malicious insiders to obtain sensitive data and for outside hackers to gain access to servers using social engineering attacks.

These growing trends are just one reason why centralized and efficient protection of sensitive data regardless of the applications being used is more important than ever.[1] Putting in place security measures that consistently protect sensitive data at the source is a critical control needed as stored data continues to proliferate and access to data expands beyond traditional boundaries. Protecting data requires a defense in depth, multi-layered approach that encompasses preventive, detective, and administrative controls.

Advanced Security option delivers two essential preventive controls covering encryption of data-at-rest and redaction of sensitive data displayed by applications.[2] These controls help protect sensitive data from being exposed directly from storage or through applications. Oracle Advanced Security Transparent Data Encryption (TDE) helps prevent attacks that attempt to bypass the database and read sensitive information from data files at the operating system level, from database backups, or from database exports.

Oracle Advanced Security Data Redaction complements TDE by reducing the risk of unauthorized data exposure in applications, redacting sensitive data in query results before the data leaves the database. This research describes TDE and Data Redaction and explains how these valuable preventive controls can work together to help secure users’ sensitive data.
1.1. Objectives of Research Study

We discuss important considerations in the development of a research thesis and in defining objectives for research.

1.1.1. Preventing Database Bypass with Encryption

Data-at-rest encryption is an important control for blocking unauthorized access to sensitive data using methods that circumvent the database. Privileged operating system accounts are just one of the vehicles used by attackers as well as insiders to gain access to sensitive information directly in physical storage. Advanced Security Transparent Data Encryption (TDE) stops attackers from bypassing the database and reading sensitive information from storage by encrypting data in the database layer.

Applications and users authenticated to the database continue to have access to application data transparently, while unauthenticated users attempting to circumvent the database are denied access to clear text data. To understand this better, consider the fact that privileged operating system users can access database tablespace files and extract sensitive data using simple shell commands.

In addition, consider the possibility of attacks that read sensitive data from lost, stolen, or improperly decommissioned disks or backups. Figure 1.1 an example of extracting customer credit card numbers directly from storage using the common Linux “strings” command and a search pattern.
1.1.2. Advanced Security Transparent Data Encryption

Transparent Data Encryption resides at an optimal layer within the database to prevent database bypass while still being transparent to applications and easy to deploy. TDE can encrypt individual application table columns or entire application tablespaces. It is transparent to applications because the encryption and decryption processes do not require any application changes, and the application users do not have to directly deal with encrypted data.[3] Most importantly, TDE’s built-in two-tier encryption key management provides full key lifecycle management, tracking the keys across their lifetime with helpful meta-data attributes, and assisted encryption key rotation, switching to a new master key with no downtime. Figure 1.2 shows how encrypting an Oracle database using TDE prevents database bypass.

Figure 1.1 Extracting customer credit card numbers from Oracle database tablespace files

Figure 1.2 Encrypting with Transparent Data Encryption to prevent database bypass
1.2. Proposed System

The proposed database system provides two critical preventive controls: i) Transparent Data Encryption encrypt data at rest to stop database bypass ii) attacks from accessing sensitive information in storage. iii) Data Redaction reduces exposure of sensitive information in applications by redacting database query results on-the-fly, according to defined policies. iv) Together these two controls form the foundation of a multi-layered, defense-in-depth approach.

1.2.1. Performance Characteristics of TDE

TDE’s cryptographic operations are extremely fast and well integrated with relational Database features. TDE leverages CPU-based hardware cryptographic acceleration available in Intel® AES-NI and Oracle SPARC T4/T5 platforms to increase performance by up to 5 xs or more. The block-level operations of TDE tablespace encryption receive an additional performance boost from database buffering and caching. Tablespace encryption integrates seamlessly with Advanced Compression, ensuring that compression occurs before encryption. Tablespace encryption also integrates with the advanced technologies in Exadata such as Exadata Hybrid Columnar Compression (EHCC) and Smart Scans, which offload certain cryptographic processing to storage cells for fast parallel execution [7].

1.3 Built-In Key Management

Key management is critical to the security of the encryption solution. Advanced Security TDE provides an out-of-the-box, two-tier key management architecture consisting of data encryption keys and a master encryption key. The data encryption keys are managed automatically and are encrypted by the master encryption key. The master encryption key is stored and managed outside of the database within an Oracle Wallet (a standards-based PKCS12 file that protects keys). Keeping the master key separate from the encrypted data mitigates attacks because both the keys and the encrypted data must be separately
compromised to gain access to clear data. The two-tier key architecture also enables rotation of master keys without having to re-encrypt all of the sensitive data. Advanced Security introduces a new dedicated SYSKM role for optional delegation to a designated user account that may run all key management operations including rotating master keys and changing the keystore password. Oracle Enterprise Manager provides a convenient graphical user interface for creating, rotating, and managing TDE master keys as shown in the figure Fig 1.3 below.

![Figure 1.3 Managing and rotating TDE master keys using Relational Database Enterprise Manager](image)

1.4 Encryption Impact for Common Operational Activities

Essential day-to-day database operational activities can potentially leak sensitive data when not performed properly, making bypass easy. Examples of these activities include database backup and restore, data movement, high-availability clustering, and replication. [4] Advanced Security TDE supports these critical database operational activities and helps ensure that the data remains encrypted. Tablespace encryption integrates with Recovery Manager (backup and restore), Oracle Data Pump (data movement), Active Data Guard (redundancy and failover), and Golden Gate (replication). TDE also
integrates with internal features of the database such as redo to prevent possible data leakage in logs

<table>
<thead>
<tr>
<th>Database Products and Technologies</th>
<th>Example Points of Integration</th>
<th>TDE Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exadata</td>
<td>Exadata Smart Scans, Exadata Hybrid Columnar Compression</td>
<td>✔</td>
</tr>
<tr>
<td>Database Compression</td>
<td>Oracle Advanced Compression</td>
<td>✔</td>
</tr>
<tr>
<td>Backup and Restore</td>
<td>Oracle Recovery Manager (RMAN), Oracle Secure Backup</td>
<td>✔</td>
</tr>
<tr>
<td>Export and Import</td>
<td>Oracle Data Pump Export and Import</td>
<td>✔</td>
</tr>
<tr>
<td>High-Availability Clusters</td>
<td>Oracle Real Application Clusters (RAC), Oracle Active Data Guard</td>
<td>✔</td>
</tr>
<tr>
<td>Storage Management</td>
<td>Oracle Automatic Storage Management (ASM)</td>
<td>✔</td>
</tr>
<tr>
<td>Database Replication</td>
<td>Oracle GoldenGate</td>
<td>✔</td>
</tr>
</tbody>
</table>

Table 1.1 Example integrations with Advanced Security TDE

This fully integrated approach to database encryption makes the solution easy to deploy in complex real-world environments, while protecting against bypass attacks that attempt to take advantage of gaps in operational processes.

1.5 Limiting Sensitive Data Exposure with Data Redaction

Privacy and compliance require a cost effective approach to managing data exposure in applications. The embrace of Smartphone and tablet devices make the issue of sensitive data exposure even more urgent as data access beyond the traditional office environment becomes commonplace.[5] Even traditional applications require a more comprehensive solution for reducing exposure to sensitive data. Take for example, a call center application with a screen that exposes customer credit card information and personally identifiable information to call center operators. Today, exposure of that information, even to valid application users, may violate privacy regulations and put the data at unnecessary risk.
1.6. Advanced Security Data Redaction

Advanced Security Data Redaction provides selective, on-the-fly redaction of sensitive data in database query results prior to display by applications so that unauthorized users cannot view the sensitive data. The stored data remains unaltered, while displayed data is transformed and redacted on-the-fly before it leaves the database. Data Redaction reduces exposure of sensitive information and helps prevent exploitation of application flaws that may disclose sensitive data in application pages. It is well suited for both new and legacy applications that need to limit exposure of sensitive data without invasive application changes.
1.7. Policies and Transformations

Advanced Security Data Redaction supports a number of different transformations that can redact all data in specified columns, preserve certain pieces of the data, or randomly generate replacement data. Examples of the supported data transformations are shown below:

Figure 1.6 Example Data Redaction transformations

Data Redaction makes the business need-to-know decision based on declarative policy conditions that utilize rich runtime contexts available from the database and from the applications themselves. Examples include user identifiers, user roles, and client IP addresses. Context information available from Oracle Application Express (APEX), Oracle Real Application Security, and Oracle Label Security also can be utilized. Redacting APEX applications is straightforward because you can create policy conditions using the application users and application identifiers that APEX automatically tracks. Multiple runtime conditions can be joined together within a Data Redaction policy for fine-grained control over when redaction occurs. The policies are stored and managed inside of the database, and they go into effect immediately upon being enabled.
1.8. Performance Characteristics of Data Redaction

High-speed performance is crucial for Data Redaction because the target databases typically will be production systems. Data needs to be transformed on-the-fly at runtime, without altering data stored on disk or in caches and buffers. Because the transformations will execute on production environments and will be repeated frequently, the performance overhead must be small. [6] One important performance characteristic of Data Redaction is that it supports only data transformations with proven high performance. These are a subset of all the possible operations that could be used to transform data in non-production environments. This specific subset avoids long running and processor intensive operations. Data Redaction also leverages performance optimizations of the Database that are only possible by being part of the database kernel. The implementation ensures that data transformations are fast in-memory computations. Policy information is cached in memory and policy expressions are evaluated only once per execution, so there is no per row performance impact.

1.9. Security Considerations

Another benefit from Data Redaction being part of the database kernel is tighter security. The implementation avoids tempering methods that plague other redaction techniques due to their dependence on proxies that can be meddled with. Additionally, Data Redaction in the kernel continues protecting when other security measures may be compromised. For example, runtime conditions in policies can narrow the impact of a SQL Injection attack by continuing to redact sensitive data even when the attack has bypassed other preventive controls in the application and database. Data Redaction also avoids obvious sources of leakage where the redaction policy could be bypassed by copying data into a new table that does not have a policy. Certain mass copy operations that touch redacted data are blocked by default, and this behavior can be overridden where necessary using a Data Redaction exempt privilege. Although Data Redaction can be used to prevent accidental viewing of sensitive data by privileged database users such as DBAs, it is intended primarily for redacting data displayed by software applications. Data Redaction does not prevent privileged users from connecting directly to the database
and running ad hoc queries that back into pieces of sensitive data (i.e. it does not stop exhaustive ad hoc queries or other inference attacks). However, note that Data Redaction is fully compatible with Oracle Database security solutions that control and monitor privileged database user access, including DBAs. It can be deployed in tandem with Oracle Database Vault or Oracle Audit Vault & Database Firewall. Data Redaction can be used with database encryption as well, and it is a great complement to TDE [7].

1.10. Easy to Deploy Data Redaction

Data Redaction can be deployed for existing applications quickly using either a command line API or Oracle Enterprise Manager. The command line API is a PL/SQL procedure that accepts protected columns, transformation types, and conditions. The Enterprise Manager provides a convenient Policy Expression Builder that enables administrators to define and apply redaction policies on existing applications. As shown below, the Policy Expression Builder dialog guides the user through creating policy conditions that use context obtained from applications, the database, the APEX framework, and other database security solutions.

![Policy Expression Builder](image)

Figure 1.7 Enterprise Manager Policy Expression Builder to create Data Redaction policies
Predefined column templates also are available in Oracle Enterprise Manager for redacting common sensitive data such as credit card numbers and U.S. Social Security numbers. Enterprise Manager Sensitive Data Discovery assists in locating columns to be redacted inside of complex application schemas.

Another reason why Data Redaction is easy to deploy is its transparency to applications and the database. For application transparency, Data Redaction supports the column data types that are frequently used by applications and various database objects including tables, views, and materialized views. Redacted values retain key characteristics of the original data such as the data type and optional formatting characters. Random redaction values are drawn from data ranges defined by the existing column data. For transparency to the database, Data Redaction avoids impacting essential database operational activities. It does not affect administrative tasks such as data movement (Data Pump) or database backup and restore (Recovery Manager). It does not interfere with database cluster configurations such as Real Application Clusters, Active Data Guard, and Golden Gate. Data Redaction does not get in the way of existing database triggers or Virtual Private Database (VPD) policies. And because Data Redaction is part of the database kernel, no separate installation is required [8].